



Interne Tabelle DDIC

Eine interne Tabelle anhand einer gegebenen DDIC-Struktur erzeugen und Ausgabe als einfache Liste

Verwendung

Die in diesem Beispiel verwendete Möglichkeit interne Tabellen und Arbeitsbereiche dynamisch zu erzeugen kann verwendet werden, wenn der Inhalt einer Datenbanktabelle ausgelesen werden soll.

So könnte man zum Beispiel ein Programm schreiben, das alle Tabellen ausliest, in denen das Datenelement „MATNR“ vor kommt. Für jede Tabelle wird dann dynamisch eine interne Tabelle generiert und mittels ASSIGN COMPONENT ‚MATNR‘ könnten dynamisch Prüfungen auf die Materialnummer erfolgen.

Coding

=== data

```
DATA gr_struct TYPE REF TO cl_abap_structdescr.  
DATA gr_table TYPE REF TO cl_abap_tabledescr.  
DATA gd_table TYPE REF TO data.  
DATA gd_line TYPE REF TO data.
```

=== field symbols

```
FIELD-SYMBOLS <workarea> TYPE ANY.  
FIELD-SYMBOLS <table> TYPE ANY TABLE.  
FIELD-SYMBOLS <value> TYPE ANY.
```

```
PARAMETERS p_table TYPE tabname DEFAULT ‚T000‘.
```

```
START-OF-SELECTION.
```

=== get structure description

```
gr_struct ?= cl_abap_typedescr=>describe_by_name( p_table ).  
*== create table description using structure description  
gr_table ?= cl_abap_tabledescr=>create( gr_struct ).
```

=== create work area

```
CREATE DATA gd_line TYPE HANDLE gr_struct.  
ASSIGN gd_line->* TO <workarea>.
```

=== create internal table

```
CREATE DATA gd_table TYPE HANDLE gr_table.  
ASSIGN gd_table->* TO <table>.
```

***== select data from database into dynamically generated internal table**

```
SELECT * FROM (p_table)  
INTO CORRESPONDING FIELDS OF TABLE <table> UP TO 20 ROWS.
```

***== loop at all table lines for output**

```
LOOP AT <table> ASSIGNING <workarea>.  
NEW-LINE.  
DO.
```

***== loop at all components of the structure**

```
ASSIGN COMPONENT sy-index OF STRUCTURE <workarea> TO <value>.  
IF sy-subrc > 0.  
EXIT.  
ELSE.
```

***== and write into list**

```
WRITE <value> COLOR COL_NORMAL.  
ENDIF.  
ENDDO.  
ENDLOOP.
```